REMARKS

Reconsideration and allowance are respectfully requested.

Initially, given the new reference being used in the rejection, the finality of this office action is improper, and should be withdrawn. Notice of this withdrawal, so that applicant can then set forth a proper supplemental amendment, is respectfully requested.

The Examiner rejects claims 2-3, 6-7, 16-18, 20-21, 29 and 47 under 35 USC 103(a) as being unpatentable over Ageenko et al. in view of Johns 6,366,289.

The Examiner rejects claims 4, 11, 19 and 54 under 35 USC 103(a) as being unpatentable over Ageenko et al., and further in view of Simms 5,586,280.

The Examiner rejects claim 14 under 35 USC 103(a) as being unpatentable over Ageenko et al., in view of Johns 6,366,289 as applied to claim 17, and further in view of Morikawa et al. 6,043,897.

Applicant traverses the above rejections.

Initially, with respect to independent claim 17, applicant objects to the characterization of either Ageenko or Johns using threads.

The Examiner has misinterpreted the term "thread" as used in this claim. The reasoning for doing so is evident: using an inappropriate interpretation makes possible an inappropriate characterization of the prior art.

The word thread, as used in computer science, is widely understood as a sequence of program instructions. The specification usage is consistent with this ordinary usage of the word "thread," and makes it clear that this is the only usage intended. For example, the specification teaches that the "C/D [compression/decompression] engine 230 being implemented using at least one processor that has the capability of operating multiple threads simultaneously." Still further, a number of different processors in parallel can be used to even more efficiently implement the C/D engine 230. No matter which implementations are used, the controller interface and the C/D engine 230 are preferably implemented as a sequence of program instructions, written in C++ or some other computer language or, alternatively, implemented in hardware.

Consistent with the ordinary usage of the word "thread" in this computer science context, processor operations operate upon data, and that is explicitly referred to in claim 1:

"each first thread includes at least one of the plurality of blocks," with the blocks being digital data. Thus, it is clear that threads are explicitly recited, as are the blocks of data associated with each thread.

In contrast to this construction, the Examiner reads "each first thread includes at least one of the plurality of blocks" as meaning that a thread is the same as or reads on "chunks". Hence, given an inappropriate construction "threads" the Examiner reaches the conclusion that "threads" are the same as chunks of data and, hence, the combination of references render claim 17 obvious.

Not only is this reasoning improper, but it is not supported by Ageenko or Johns. The Examiner admits Ageenko does not teach threads. Johns specifically teaches -- consistent with ordinary usage in the art -- that a thread is related to related to a "process" (i.e. sequence of instructions) at column 15, lines 35-39. Further, John's teaches that this process (or thread) operates upon "chunks." As such, Johns refers to a thread in the conventional manner known by one of ordinary skill in the art. Given Johns teaching, and differentiation between the use of the word thread and chunks, at best Johns teaches using two threads (a background thread and a foreground thread), with the background thread having chunks of data associated with it. Johns does not teach a plurality of threads with each thread including a plurality of blocks, and then operating on each of the plurality of threads.

The Examiner, however, refers to a different portion of Johns for a teaching of thread - though this is internally inconsistent with Johns. The cited passage, column 9, lines 29-45, teaches that a single "decompressor" can be used to implement two different decompressors, such that a single decompressor can serve two different clients. This has nothing to do with "threads", and the Examiner continued attempt to force this is just wrong. Applicant notes that the Examiner continues, in successive amendments, to try to use different teachings of Johns in different manners to suggest that Johns teaches the present invention. Johns does not, and as such the continued reliance on Johns is improper. Generally stated, Johns teaches that managing subregion called a chunk that contains compressed and uncompressed pixel memory. In contrast, Appellant's claim 17 recites "creating a plurality of second threads that include at least one of the plurality of compressed blocks . . ." Thus, managing a chunk that has compressed and uncompressed pixel memory, as taught by Johns, is not the same as creating second threads that each include, among other things, previously compressed blocks, as set forth in Appellant's

claim 17. Consequently, the Examiner's conclusion that this element of Appellant's claim is anticipated by Johns is a mere conclusion and incorrect.

Further, claim 17 is directed to a method of further compressing already compressed blocks. This is not taught or suggested by Ageenko. Different clusters are each compressed once, as in typical in the art. There is not, however, any teaching whatsoever of compressing blocks, and then further compressing the already compressed blocks again. As such, Ageenko does not teach or suggest this element.

Claim 20 highlights the above distinction even more clearly, with the language "during the step of operating upon each of the plurality of second threads, a same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block."

Other dependent claims also contain allowable subject matter. For example, with respect to claims 3 and 18, Ageenko does not teach threads as admitted by the Examiner, yet the Examiner does not even refer to a teaching in Johns to support. There is not such support. As previously argued in previous communications, Johns does not teach or disclose the concept of threads that include data in light of independently operating upon each of the plurality of first threads. The portions of Johns *previously* (but not in this action) cited by the Examiner for this proposition are directed toward two entirely different concepts, those being (1) that "the compression process can treat each coefficient independently" and (2) "the compression methods described above compress images in independent blocks" within the context of data compression. While each uses the word "independently," as does claim 3, this word usage does not teach or suggest independently operating upon a plurality of threads to obtain compressed data associated with each different thread.

Further, with respect to claims 6 and 21, this rejection is inappropriate in light of the misinterpretation of threads. And, as previously asserted in previous communications, the Examiner has indicated that "'combining the plurality of compressed first threads' is read on 'compressed chunks are linked together in a linked list format'." In addition to the interpretational differences argued by Applicant above with respect to "thread," this rejection further illustrates the Examiner's error relating to the threads. In particular, the claim requires "combining" compressed blocks from the threads after the compression operation using the plurality of threads, as recited. In contrast, the Examiner points to an operation in Johns that

creates a linked list of various chunks of data, but these various chunks of data have not been previously separated and associated with various threads, and then operated upon as recited. Thus, claims 6 and 21 contains allowable subject matter.

Further, the Examiner rejected claim 7 based on the same misinterpretation and misapplication of Johns. If properly interpreted, claim 7 is patentably distinct.

Even if the Examiner's interpretation is used, however, claim 7 is allowable. Appellant's claim 7 recites that "the step of creating the plurality of first threads includes the step of associating each of the plurality of blocks of digital data with one of the plurality of first threads." This relates to a distinct association between each thread and a block of data. In contrast, the teaching of Johns, which the Examiner relies on to reject claim 7, specifically states that the VFB controller "computes the address of the compressed block control data associated with the chunk and reads this control data from the memory device where it resides." See Johns, col. 7 at lines 62-66. It is clear that Johns teaches an association between the compressed block control data and the chunk; in other words data is being associated with data. Further, however, this association is being performed *after* compression has already taken place. In contrast, as recited, in the step of creating the first plurality of threads, data is being associated with different threads *prior* to it being compressed, and as such Johns does not provide teaching that render this claim unpatentable.

With respect to claims 4, 19 and 54, the Examiner adds Simms to the rejection -- the same reference used previously for this teaching. As previously argued, the Examiner's rejection of claim 4 further demonstrates Examiner's misinterpretation of Appellant's invention as set out in claim 4. More specifically, the Examiner has indicated that Johns does not teach "threads are independently operated upon in parallel." Additionally, Simms does not teach operating upon threads in parallel. The Examiner provides no reason for motivation to combine these references (although the Examiner's previous assertion that the motivation to combine in order to establish obviousness is based upon "knowledge generally available in the art" because that "would increase efficiency and reduce operation time" is wholly speculative and unsupported. The mere fact that such parallel operations could potentially increase efficiency and reduce operation time is not sufficient because such as a conclusion does not explain the increase complexity and solution for potential problems that may cause malfunction. Thus, on the established record

there is no support for the Examiner's assertion that such a motivation would have been part of generally available knowledge as this only takes into account the end result and not how that end result is achieved.)

Furthermore, the Examiner has again relied upon the misinterpretation that threads are the same as data to conclude that Simms teaches "certain ones of first threads are independently operated upon in parallel." However, the portion of Simms that the Examiner relies upon is *unrelated* to threads; the portion of Simms relates specifically to data. It is clear, as the Examiner has referenced, that Simms teaches that "[f]or data which is being compressed, in parallel with organization of compressed data into groups . . ." See Simms, col. 17, lines 16-20. Thus, based on the foregoing, Simms teaches that compression of data can occur while the compressed data is organized into groups. However, this is unrelated to the concept of independently operation upon certain first threads in parallel; compression of data in parallel with organization of compressed data into groups is not the same as independently operation upon certain threads in parallel. It is also clear, again as the Examiner has referenced, that Simms teaches in "parallel with the generation of the main data blocks, 35-byte sub-data blocks are also generated . . ." See Simms, col. 19, lines 27-28. However, it is unclear how generation of data in parallel is related to independently operating upon certain threads in parallel. The only way that the Examiner can reach that conclusion that modifying "Johns by the teaching of Simms" would have been obvious is to first conclude that a thread is the same data. This is a clear misinterpretation of the Appellants' claims 4, 19, and 54, which each state that threads include data and threads are independently operated upon in parallel. And applicant notes that claim 54 requires *both* the first and second threads to be operated upon in parallel -- a suggestion nowhere found in any of these references.

With respect to claim 11, Appellant's claim 11 is not obvious in light of Ageenko andJohns as modified by Simms. Claim 11 teaches the

"step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type."

Additionally, neither Ageenko, Johns nor Simms contain "threads", for the reasons recited, the Examiner further asserts that Simms teaches

"the step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type (see column 7, lines 11-16)."

In short the Examiner has merely recited Appellant's claim language and provide a citation to Simms and concluded that this would have been obvious because it would "enable the system for categorize data into blocks of data with common characteristics among the data items." The cited portion of Simms, however, does not support Examiner's rejection given that the cited portion specifically states that

"entries in the block access table each comprise a FLAG entry indicating the type of the entry and a COUNT entry indicating its value. The FLAG field is 8 bits and the COUNT filed is 24 bits. The bits in the FLAG field have the following significance . . ." Accordingly, the subject matter of claim 11 is patentably distinct.

With respect to claim 14, the Examiner adds Simms to the rejection -- the same reference used previously for this teaching. As previously argued, the Examiner's assertion that Morikawa teaches "the step of predicting and estimated compression time (see column 2, lines 14-18) and estimated compression amount for each block (see column 5, lines 57-63)" is based upon a misinterpretation of Morikawa.

Initially, Morikawa is combined with Ageenko and Johns in a manner where there would not have been motivation to do so. Morikawa is a specific technique used in a device used to print on paper an electronic document, which is not the concern of Ageenko or Johns at all. As such, one of ordinary skill in the art would not have attempted the combination suggested by the Examiner, and this illustrates that inappropriate hindsight is being used.

Assuming arguendo that the art did teach the modification of Ageenko and Johns by Morikawa, such a combination still does not teach, disclose, or suggest Appellant's invention as set forth in claim 14. In particular, Morikawa, however, measures the compression time for one block in order to estimate the time for compressing all of the blocks, which is not the same as "estimating compression time . . . for each block" as set forth in Appellant's claim 14.

Similarly, Morikawa teaches actually measuring and storing information related to "size of compressed image and measured compression time" without any reference to estimating

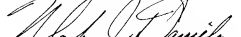
compression amount. See id, col. 5, lines 45-63. Thus, measuring and storing information related to actual "size of compressed image", as taught by Morikawa, is not the same as "estimating compression amount for each block" as set forth in Appellant's claim 14. Thus, the Examiner's rejection of claim 14 is improper.

All objections and rejections having been addressed, and in view of the foregoing, the claims are believed to be in form for allowance, and such action is hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, he is kindly requested to contact the undersigned at the telephone number listed below.

Date: April 26, 2006

Respectfully submitted,

PILLSBURY WINTHROP SHAW PITTMAN LLP



Mark J. Danielson, Reg. No. 40,580 for
David A. Jakopin, Reg. No. 32995
Tel. No. 650-233-4790

Customer No. 27,498
2475 Hanover Street
Palo Alto, CA 94304-1114
(650) 233-4500